Theory of Computing Systems (2022) 66:432–453
https://doi.org/10.1007/s00224-022-10073-0

On the Parameterized Complexity of the Expected Coverage Problem

Fedor V. Fomin[1] · Vijayaragunathan Ramamoorthi[2]

Accepted: 28 January 2022 / Published online: 10 March 2022

Abstract

The MAXIMUM COVERING LOCATION PROBLEM (MCLP) is a well-studied problem in the field of operations research. Given a network with positive or negative demands on the nodes, a positive integer k, the MCLP seeks to find k potential facility centers in the network such that the neighborhood coverage is maximized. We study the variant of MCLP where edges of the network are subject to random failures due to some disruptive events. One of the popular models capturing the unreliable nature of the facility location is the linear reliability ordering (LRO) model. In this model, with every edge e of the network, we associate its survival probability $0 \leq p_e \leq 1$, or equivalently, its failure probability $1 - p_e$. The failure correlation in LRO is the following: If an edge e fails then every edge e' with $p_{e'} \leq p_e$ surely fails. The task is to identify the positions of k facilities that maximize the *expected* coverage. We refer to this problem as EXPECTED COVERAGE problem. We study the EXPECTED COVERAGE problem from the parameterized complexity perspective and obtain the following results. 1. For the parameter pathwidth, we show that the EXPECTED COVERAGE problem is W[1]-hard. We find this result a bit surprising, because the variant of the problem with non-negative demands is fixed-parameter tractable (FPT) parameterized by the treewidth of the input graph. 2. We complement the lower bound by the proof that EXPECTED COVERAGE is FPT being parameterized by the treewidth and the maximum vertex degree. We give an algorithm that solves the problem in time $2^{\mathcal{O}(\mathbf{tw} \log \Delta)} n^{\mathcal{O}(1)}$, where $\mathbf{tw}$ is the treewidth, Δ is the maximum vertex degree, and n the number of vertices of the input graph. In particular, since $\Delta \leq n$, it means the problem is solvable in time $n^{\mathcal{O}(\mathbf{tw})}$, that is, is in XP parameterized by treewidth.

This work was done while the second author was visiting University of Bergen, Bergen, Norway supported by the Norwegian Research Council (NFR) MULTIVAL project.

✉ Vijayaragunathan Ramamoorthi
vijayr@cse.iitm.ac.in

Fedor V. Fomin
fedor.fomin@ii.uib.no

[1] Department of Informatics, University of Bergen, 5020 Bergen, Norway

[2] Department of Computer Science and Engineering, IIT Madras, Chennai, India

Keywords Facility location · Treewidth · W[1]-hard · Pathwidth · Negative demands

1 Introduction

The Maximum Covering Location Problem (MCLP) is a well-studied problem in the field of operations research [8]. Given a network with demands on the nodes, a positive integer budget k, the MCLP asks to find k potential facility centers in the network such that the neighborhood coverage is maximized. We are interested in investigating the unreliable nature of the MCLP. Unreliability is introduced by associating survival probabilities on the edges of the input network. The notion of unreliability is used in disaster management, surviving network design and influence maximization. Assume that the network is subjected to a disaster event. During the course of disaster, some link may become non-functional. This yields a structural change in the underlying graph of the network. The resulting graph is an edge-induced subgraph of the original graph. In certain cases, the resulting graph can have multiple connected components. The real challenge is to place a limited number of potential facility centers a priori such that the expected coverage after an event of disaster is maximized. See [9, 13–15, 19, 32] for further references on unreliable MCLP.

In this paper, we study the following model of the MCLP with edge failures. Let $G = (V, E, w)$ be a vertex weighted underlying graph of the MCLP. On each edge $e \in E$, let $p_e > 0$ be the survival probability associated with e such that the edge e can survive in the network with probability p_e. Under the assumption that edges fail independently, the input graph can be rendered into one of 2^m edge subgraphs called *realization*, where m is number of edges in the graph. Each realization will have a non-zero probability of occurrence. Since the number of realizations is exponential and many of them occur with close to zero probability, Hassin et al. [26, 27] formulated a dependency model for edge failure in unreliable facility networks called *linear reliability ordering* (LRO). In LRO model, for each pair of edges $e \neq e' \in E$, $p(e) \neq p(e')$, and for any pair of edges e_i and e_j with $p_{e_i} > p_{e_j}$, the $\Pr[e_j \text{ fails } \mid e_i \text{ fails }] = 1$. More precisely, if an edge e fails then every edge e' with $p_{e'} < p_e$ surely fails. The LRO model is defined on graphs with distinct edge probabilities. It is clear that, in this model, we have exactly $m + 1$ edge subgraphs. We consider the LRO model with a relaxation that the edges can have the same probability. If the probabilities of two edges are the same, then either both or neither will survive. In this case, the number of subgraph realization will be at most $m + 1$.

While in most articles dealing with maximum coverage problems the weights are assumed to be positive, there are situations when the weights can be negative. Such mixed-weight coverage problems are useful for modeling situations when some of the demand nodes are obnoxious and their inclusion in the coverage area may be detrimental [4, 5]. Nodes with a negative demand are nodes we do not wish to cover. If a node has negative demand, then we wish to cover as little as possible. For example, opening a new facility (grocery store) close to many positive weighted modes (customers) seems as an excellent opportunity but the proximity of a big supermarket (a neighbor with negative weight) could decrease the expected profit.

Problem Statement Let $G = (V, E)$ be a vertex-weighted undirected graph with a weight function $w : V \to \mathbb{R}$ and a probability function $p : E \to \mathbb{Q}_{[0,1]}$, and k be a positive integer. Assume that the edges are ordered using p in descending order. That is, $p_1 > p_2 > \cdots > p_m$. In the LRO model, let $G_0 \preceq G_1 \preceq \cdots \preceq G_m$ be the linear ordering of the realizations of G, that occur with probability $P(G_i)$ for $0 \le i \le m$. The value of $P(G_i)$ can be written as follows:

$$P(G_i) = \begin{cases} 1 - p_1 & \text{if } i = 0, \\ p_m & \text{if } i = m, \\ p_i - p_{i+1} & \text{otherwise.} \end{cases}$$

The Expected Coverage problem asks to find a k-sized vertex set S such that the expected coverage by S on the distribution $\{G_i \mid 0 \le i \le m\}$ is maximized. We use the expected coverage function $\mathcal{C}$ defined by Narayanaswamy et al. [33]. Given a pair of sets $S, T \subseteq V$, the expected coverage of T by S is

$$\mathcal{C}(T, S) = \sum_{i=0}^{m} \Big(P(G_i) \sum_{v \in N_{G_i}[S] \cap T} w(v) \Big).$$

Further, if S or T is a singleton set, we just write the element of the set instead of the set notation. An instance of the optimization version of the Expected coverage problem is denoted by the tuple (G, w, p, k). The decision version of the problem is defined as follows.

Expected Coverage
Instance: A graph $G = (V, E)$, $w : V \to \mathbb{Q}$, $p : E \to \mathbb{Q}_{[0,1]}$, a positive integer k and value of coverage $t \in \mathbb{R}$
Decide: Is there a set $F \subseteq V$ of size at most k such that $\mathcal{C}(V, F) \ge t$.

An instance of the decision version of the Expected Coverage problem is denoted by the tuple (G, w, p, k, t).

Related Works The facility location problems can take many forms, depending on the objective function. In the most facility location problems, the objective function focuses on comforting the clients. For example, in the k-center problem, the goal is minimizing the maximum distance of each client from its nearest facility center [7]. The facility location problem has received a good deal of attention in the parameterized perspective [1, 6, 20, 21].

The MCLP with edge failure is studied with various constraints. Eiselt et al. [19] considered the problem with a single edge failure. In this case, exactly one edge would have failed after a disaster and the objective is to place k facility centers such that the expected weight of non-covered vertices is minimized. If the number of facility centers is $k = 1$, and the facility center can cover all the vertices in the connected component, then the problem is studied as Most Reliable Source (MRS) problem. In this problem, the edges fail independently. The MRS problem has received a good deal of attention in literature [9, 13, 15, 32]. Hassin et al. [26] studied the problem with edge failure follows LRO failure model. The problem is referred as MAX-EXP-COVER-R problem. An additional input radius of coverage R is also given such that any facility center can cover a vertex at distance at most R. The MAX-EXP-COVER-R problem is shown to be NP-hard even when $R = 1$. When $R = \infty$ (it is sufficient to say $R > n$), the problem is polynomial time solvable [26].

In the BUDGETED DOMINATING SET problem, we are given a graph G and a positive integer k, and asked to find a set of at most k vertices S maximizing the value $w(N[S])$ in G. Set theoretic version of the BDS problem is studied as budgeted maximum coverage in [28, 29]. The EXPECTED COVERAGE problem can be viewed as a generalization of the BDS problem. When we have probability 1 on all the edges, then both these problems are the same. The BDS problem generalizes PARTIAL DOMINATING SET (PDS) problem, where one seeks a set of size at most k vertices dominating at least t vertices [31]. Of course, all these problems also generalize the fundamental DOMINATING SET problem, where the task is to find a set of at most k vertices dominating all remaining vertices of the graph.

The DOMINATING SET problem parameterized by k (solution size) on general graphs is W[2]-hard [16]. However, on planar graphs it is FPT [25]. Moreover, on planar, and more generally on H-minor-free graphs it is solvable in sub-exponential time [3, 11]. It also admits a linear kernel on planar graphs, H-minor-free graphs and graphs of bounded expansion [2, 18, 23, 24, 34]. Sub-exponential parameterized algorithm for the PDS problem on planar graphs, and more generally, apex-minor-free graphs, was given in [22].

On graphs of bounded treewidth, the classical dynamic programming, see e.g. [10], shows that the DOMINATING SET problem is FPT parameterized by the treewidth of an input graph. The FPT algorithm for the DOMINATING SET problem can be adapted to solve the BDS problem in FPT time. Further, when we have mixed vertex weights on the BDS problem, the above modified algorithm will work. Narayanaswamy et al. [33] gave an FPT algorithm parameterized by treewidth of the input graph to solve the EXPECTED COVERAGE problem with *non-negative* weights.[1]

Our Results Since the EXPECTED COVERAGE problem (with mixed-weights) generalizes both the BDS problem and the EXPECTED COVERAGE problem with non-negative weights, it is also natural to ask what algorithmic results for these problems can be extended to the EXPECTED COVERAGE problem. We obtain the following results.

1. For the parameter pathwidth, we show that the EXPECTED COVERAGE problem is W[1]-hard. Moreover, the problem remains W[1]-hard for any combination of parameters pathwidth **pw**, solution size k and value of coverage t. This is interesting because as it was shown by Narayanaswamy et al. [33], the variant of the problem with only non-negative weight is FPT parameterized by the treewidth. Thus the results for non-negative weights cannot be (unless FPT=W[1]) extended to the mixed-weight model.

[1] Narayanaswamy et al. [33] called this problem MAX-EXP-COVER-1-LRO.

2. We complement the lower bound by the proof that EXPECTED COVERAGE is FPT being parameterized by the treewidth and the maximum vertex degree. We give an algorithm that solves the problem in time $2^{\mathcal{O}(\mathbf{tw} \log \Delta)} n^{\mathcal{O}(1)}$, where $\mathbf{tw}$ is the treewidth, Δ is the maximum vertex degree, and n the number of vertices of the input graph. In particular, since $\Delta \leq n$, it means the problem is solvable in time $n^{\mathcal{O}(\mathbf{tw})}$, that is, is in XP parameterized by treewidth.

2 Preliminaries

We recall in this section some notations and definitions used throughout this article. For a positive integer x, by $[x]$ we mean the set $\{1, \ldots, x\}$. Let $\mathbb{Q}_{[0,1]}$ denote the set of all rational numbers between 0 and 1. Let $G = (V, E)$ be a simple and undirected graph with vertex set V and edge set E. Let $|V| = n$ and $|E| = m$. For any set $S \subseteq V$, by $G[S]$ we mean the subgraph of G induced by S, and by $G - S$ we mean $G[V \setminus S]$. For each vertex $u \in V$, let $deg(u)$ denote the degree of u in G. For each vertex $u \in V$, let $N(u)$ denote the open neighborhood of u. For any set $S \subseteq V$, open neighborhood of the set S in G is denoted by $N(S)$, that is, $N(S) = \cup_{u \in S} N(u) \setminus S$. Similarly, for a vertex $u \in V$ let $N[u]$ denote the closed neighborhood of u. For any set $S \subseteq V$, $N[S] = \cup_{u \in S} N[u]$. Other than this, we follow standard graph theoretic notations based on Diestel [12].

A *tree decomposition* of an undirected graph $G = (V, E)$ is a pair (T, X) where T is a tree whose vertices are called *nodes* and $X = \{X_{\mathbf{i}} \subseteq V \mid \mathbf{i} \in V(\mathsf{T})\}$ such that

1. for each vertex $u \in V$, there is a node $\mathbf{i} \in V(\mathsf{T})$ such that $u \in X_{\mathbf{i}}$,
2. for each edge $uv \in E$, there is a node $\mathbf{i} \in V(\mathsf{T})$ such that $u, v \in X_{\mathbf{i}}$, and
3. for each vertex $v \in V$, the set $\{\mathbf{i} \in V(\mathsf{T}) \mid v \in X_{\mathbf{i}}\}$ forms a subtree of T.

The *width* of a tree decomposition (T, X) equals $max_{\mathbf{i} \in V(\mathsf{T})} |X_{\mathbf{i}}| - 1$. The treewidth of a graph G is the minimum width over all tree decompositions of G. For a node $\mathbf{i} \in V(\mathsf{T})$, let $\mathsf{T}_{\mathbf{i}}$ be the subtree rooted at $\mathbf{i}$ and $X_{\mathbf{i}}{}^{+} = \cup_{\mathbf{j} \in V(\mathsf{T}_{\mathbf{i}})} \{X_{\mathbf{j}}\}$. The graph induced by the vertices $X_{\mathbf{i}}{}^{+}$ is $G[X_{\mathbf{i}}{}^{+}]$ and it is denoted by $G_{\mathbf{i}}$. A tree decomposition (T, X) is said to be a *path decomposition* if T is a path. The *pathwidth* of a graph G is minimum width over all possible path decompositions of G. Let $\mathbf{pw}(G)$ and $\mathbf{tw}(G)$ denote the pathwidth and treewidth of the graph G, respectively.

We give a dynamic programming algorithm working on a so-called *nice tree decomposition* of the input graph G. A tree decomposition (T, X) is a *nice tree decomposition* if T is rooted by a node $\mathbf{r}$ with $X_{\mathbf{r}} = \emptyset$ and every node in T is either an *insert node*, *forget node*, *join node* or *leaf node*. Thereby, a node $\mathbf{i} \in V(\mathsf{T})$ is an insert node if $\mathbf{i}$ has exactly one child $\mathbf{j}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} \cup \{v\}$ for some $v \notin X_{\mathbf{j}}$; it is a forget node if $\mathbf{i}$ has exactly one child $\mathbf{j}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} \setminus \{v\}$ for some $v \in X_{\mathbf{j}}$; it is a join node if $\mathbf{i}$ has exactly two children $\mathbf{j}$ and $\mathbf{h}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} = X_{\mathbf{h}}$; and it is a leaf node if $X_{\mathbf{i}} = \emptyset$. Given a tree decomposition of width $\mathbf{tw}$, a nice tree decomposition of width $\mathbf{tw}$ can be obtained in linear time [30]. For a node $i \in V(\mathsf{T})$, let $\mathsf{T}_{\mathbf{i}}$ be a subtree rooted at $\mathbf{i}$ and $X_{\mathbf{i}}{}^{+} = \cup_{\mathbf{j} \in V(\mathsf{T}_{\mathbf{i}})} \{X_{\mathbf{j}}\}$.

We will also use the parameters *vertex cover*, *feedback vertex set* and *distance to star forest* of a graph G. For a graph G, by $\mathbf{vc}(G)$ we mean the size of minimum vertex set whose deletion leaves the graph edgeless, and by $\mathbf{fvs}(G)$ we mean the size of the minimum vertex set whose deletion leaves the graph acyclic. In this article, we consider that the trivial graph structure is a forest that consists of only star trees. Then, for a graph G, by $\mathbf{dsf}(G)$ we mean the size of the minimum vertex set whose deletion leaves the graph disjoint union of star trees (a forest of stars). For all these structural properties, we will omit G if it is clear from the context.

We refer to the recent books of Cygan et al. [10] and Downey and Fellows [17] for detailed introductions to parameterized complexity.

3 Parameterized Intractability: The Expected Coverage problem is W[1]-hard for the Parameter Pathwidth

In this section, we show that the Expected Coverage problem is W[1]-hard parameterized by the pathwidth. We reduce from the Multi-Colored Clique problem which is defined as follows. Given a k-partite graph $G = (V, E)$ where $V = (V_1, V_2, \ldots, V_k)$, and a positive integer k, the Multi-Colored Clique problem seeks to decide whether there exists a k-clique with exactly one vertex from each part.

Theorem 1 (Cygan at al. [10]) *The* Multi-Colored Clique *problem is W[1]-complete for the parameter k.*

For each $1 \leq i < j \leq k$, let $E_{i,j} \subseteq E$ be the set of all edges where one end vertex is in V_i and another one is in V_j. That is, $E_{i,j} = \{xy \in E \mid x \in V_i \wedge y \in V_j\}$.

3.1 Construction

Given an instance (G, k) of the Multi-Colored Clique problem, we construct an instance (H, w, p, k', t') of the Expected Coverage problem where $k' = k + \binom{k}{2}$, $t' = k^4 + k^3 - k^2 + k$ and $\mathbf{pw}(H)$ is $\mathcal{O}(k^2)$. Now we describe the construction of the graph H, the function $w : V(H) \to \mathbb{Q}$ and the probability function $p : E(H) \to \mathbb{Q}_{[0,1]}$.

For each $i \in [k]$, we construct a *vertex-partition gadget H_i* corresponding to the vertex partition V_i as follows. For each vertex $u \in V_i$, add a vertex a_u with $w(a_u) = 0$ in the gadget H_i. Add two more vertices t_i with $w(t_i) = k^2$, and q_i with $w(q_i) = k^2$ to the gadget H_i. For each vertex $u \in V_i$, the vertex a_u is made adjacent to the vertices t_i and q_i. For each edge $e \in E(H_i)$, we define the survival probability $p(e) = 1$. Thus, the gadget H_i has $|V_i| + 2$ vertices and $2|V_i|$ edges.

For each $1 \leq i < j \leq k$, we construct an *edge-partition gadget $H_{i,j}$* corresponding to the edge partition $E_{i,j}$ as follows. For each edge $e \in E_{i,j}$, add a vertex a_e with $w(a_e) = 0$ in the gadget $H_{i,j}$. Add two more vertices $t_{i,j}$ and $q_{i,j}$ with $w(t_{i,j}) = k^2 = w(q_{i,j}) = k^2$ to the gadget $H_{i,j}$. For each edge $e \in E_{i,j}$, the vertex a_e is made adjacent to the vertices $t_{i,j}$ and $q_{i,j}$. For each edge $e \in E(H_{i,j})$, we define the survival probability $p(e) = 1$. Thus, the gadget H_i has $|E_{i,j}| + 2$ vertices and $2|E_{i,j}|$ edges.

Next, we introduce *connector vertices* to connect the edge-partition gadgets and vertex-partition gadgets. Let $R = \{s^i_{i,j}, s^j_{i,j}, r^i_{i,j}, r^j_{i,j} \mid 1 \leq i < j \leq k\}$ be the connector vertices. For each vertex $x \in R$, we define $w(x) = -1$. To establish the edges between the gadgets and the connector vertices, we define a probability function $z : V \to \mathbb{Q}_{[0,1]}$ such that for any two vertices $x, y \in V$ with $x \neq y$, $z(x) \neq z(y)$. For each $i \in [k]$, the gadget H_i is connected to the set R as follows. For each vertex $u \in V_i$ and for each $j \in [k]$ with $j \neq i$,

- if $i < j$ then the vertex $a_u \in H_i$ is made adjacent to the vertices $s^i_{i,j}$ and $r^i_{i,j}$ with survival probabilities $z(u)$ and $1 - z(u)$, respectively, and
- if $j < i$ then the vertex $a_u \in H_i$ is made adjacent to the vertices $s^i_{j,i}$ and $r^i_{j,i}$ with survival probabilities $z(u)$ and $1 - z(u)$, respectively.

For $1 \leq i < j \leq k$, the gadget $H_{i,j}$ is connected to the set R as follows. For each edge $e = xy \in E_{i,j}$ with $x \in V_i$ and $y \in V_j$, the vertex a_e is made adjacent to the vertices $s^i_{i,j}, r^i_{i,j}, s^j_{i,j}$ and $r^j_{i,j}$ with survival probabilities $z(x)$, $1 - z(x)$, $z(y)$ and $1 - z(y)$, respectively. An illustration of a vertex-partition gadget and an edge-partition gadget connected to the connector vertices is given in Fig. 1. For clarity, we denote the vertices a_u and a_e in $V(H)$ for each $u \in V$ and $e \in E$, as *selection* vertices. Thus, the graph H is constructed with $N = n + m + 3k^2 - k$ vertices and $M = 2kn + 6m$ edges.

Lemma 1 *For each $i \in k$, the pathwidth of the gadget H_i is two.*

Proof We observe that the removal of the vertex t_i from H_i results a star tree. It is known that the pathwidth of a star tree is one. Let $(\mathsf{T}, \mathcal{X})$ be a path decomposition of

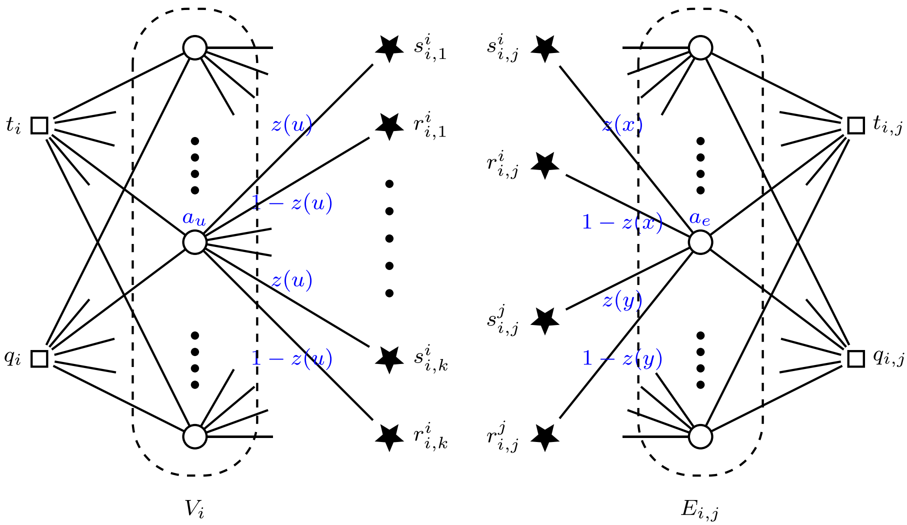

Fig. 1 Gadgets for a partition V_i and $E_{i,j}$ for some $i \neq j$ are given. Star shaped vertices are connector vertices. The selection vertices are represented by circle shape. Let $a_e \in V(H_{i,j})$ be the vertex illustrated for some edge $e = xy \in E_{i,j}$ such that $x \in V_i$ and $y \in V_j$

the graph $H_i - \{t_i\}$. Thus, adding t_i into all bags of $(\mathsf{T}, \mathcal{X})$ gives a path decomposition of h_i with pathwidth two. □

Similarly, for each $1 \le i < j \le k$, $\mathbf{pw}(H_{i,j}) = 2$. We bound some structural properties of the graph H in the following lemma.

Lemma 2 *Some structural properties of the graph H are as follows:*

(a) $\boldsymbol{pw}(H) \le 4\binom{k}{2} + 2$,
(b) $\boldsymbol{vc}(H) \le 3k^2 - k$,
(c) $\boldsymbol{fvs}(H) \le 5\binom{k}{2} + k$, *and*
(d) $\boldsymbol{dsf}(H) \le 3k^2 - k$.

Proof Consider the following two sets $T = \bigcup_{i=1}^{k}\{t_i\} \cup \bigcup_{1\le i<j\le k}\{t_{i,j}\}$ and $Q = \bigcup_{i=1}^{k}\{q_i\} \cup \bigcup_{1\le i<j\le k}\{q_{i,j}\}$. Recall that $R = \{s^i_{i,j}, s^j_{i,j}, r^i_{i,j}, r^j_{i,j} \mid 1 \le i < j \le k\}$ denotes the connector vertices. We prove each of the structural parameters mentioned above as follows.

(a) If we remove R from the graph H, then the resulting graph is a collection of disjoint vertex-partition gadgets and edge-partition gadgets. From Lemma 1, the pathwidth of a gadget is two. Let $(\mathsf{T}, \mathcal{X})$ be a path decomposition of the graph $H - R$ with pathwidth two. Thus, adding the vertex set R to all bags of $(\mathsf{T}, \mathcal{X})$ gives a path decomposition for the graph H with pathwidth at most $|R| + 2 = 4\binom{k}{2} + 2$.
(b) If we remove the set $T \cup Q \cup R$ from the graph H, then the resulting graph is edgeless. Thus, the set $T \cup Q \cup R$ is a vertex cover for H. Therefore, $\mathbf{vc}(H) \le |T| + |Q| + |R| = 3k^2 - k$.
(c) If we remove the set $T \cup R$ from the graph H, then the resulting graph is a forest. Thus, the set $T \cup R$ is a feedback vertex set for H. Therefore, $\mathbf{fvs}(H) \le |T| + |R| = 5\binom{k}{2} + k$.
(d) If we remove the set $T \cup R$ from the graph H, then the resulting graph is a forest that consists of only star trees. Therefore, $\mathbf{dsf}(H) \le |T| + |R| = 5\binom{k}{2} + k$.

□

3.2 Properties of a feasible solution for the instance (H, w, p, k', t') of the Expected Coverage problem

Let $S \subseteq V(H)$ be a feasible solution to the instance (H, w, p, k', t') of the Expected Coverage problem. That is, $|S| = k'$ and $\mathcal{C}(V(H), S) \ge t' = k^4 + k^3 - k^2 + k$. Observe that any vertex in H can achieve an expected coverage of value at most $2k^2$. In particular, for each $u \in V$ (or $e \in E$), the selection vertex a_u (or a_e) can achieve an expected coverage of value at most $2k^2$. If a vertex $u \in V(H)$ is not a selection vertex, then u can achieve an expected coverage of value at most k^2. In the following lemma, we show that the set S consists of only selection vertices.

Lemma 3 *Every vertex in the set S is a selection vertex.*

Proof We prove this by contradiction. Assume that there exists a vertex $u \in S$ where u is not a selection vertex. We know that $\mathcal{C}(V(H), u) \leq k^2$. Then, the expected coverage by the set S is given as follows:

$$\begin{aligned}\mathcal{C}(V(H), S) &\leq \mathcal{C}(V(H), S \setminus \{u\}) + \mathcal{C}(V(H), u) \leq (k' - 1)(2k^2) + k^2 \\ &= (k^2 + k - 2)(k^2) + k^2 = k^4 + k^3 - 2k^2 + k^2 \\ &= k^4 + k^3 - k^2 < t' = k^4 + k^3 - k^2 + k.\end{aligned}$$

This contradicts the feasibility of the set S. Therefore, every vertex in the set S is a selection vertex. □

Then, we show that the set S has a non-empty intersection with each gadget in the graph H.

Lemma 4 *For each $i \in [k]$, $|S \cap V(H_i)| = 1$ and, for each $1 \leq i < j \leq k$, $|S \cap V(H_{i,j})| = 1$.*

Proof By construction of the graph H, the vertex-partition gadgets and edge-partition gadgets are disjoint and connected through connector vertices. By contradiction, assume that there exists a gadget with no vertex from the gadget is in S. Since there are $\binom{k}{2} + k$ gadgets, at least one gadget should have two vertices from the set S. For any gadget, the expected coverage contribution by the vertices in the gadget is at most $2k^2$ even if the gadget has more than one vertex from S. Then we have the following:

$$\begin{aligned}\mathcal{C}(V(H), S) &\leq \mathcal{C}(V(H) \setminus R, S) \leq (k' - 1)(2k^2) = (k^2 + k)(k^2) - 2k^2 \\ &= k^4 + k^3 - 2k^2 < t'.\end{aligned}$$

This contradicts the feasibility of the set S. Therefore, the set S has one selection vertex in each gadget. □

The Lemmas 3 and 4 together state that for each $i \in [k]$, there exists a vertex $v \in V_i$ such that $a_v \in S$, and for each $1 \leq i < j \leq k$, there exists an edge $e \in E_{i,j}$ such that $a_e \in S$.

Lemma 5 *For each $1 \leq i < j \leq k$, let a_u and a_{xy} be the selection vertices in the set S for some $u, x \in V_i$ and $y \in V_j$. Then, the expected coverage of the vertices $\{s^i_{i,j}, r^i_{i,j}\}$ by $\{a_u, a_{xy}\}$ is given as follows.*

1. $\mathcal{C}(s^i_{i,j}, \{a_u, a_{xy}\}) = -(\max(z(v), z(x)))$.
2. $\mathcal{C}(r^i_{i,j}, \{a_u, a_{xy}\}) = -(\max(1 - z(v), 1 - z(x)))$.

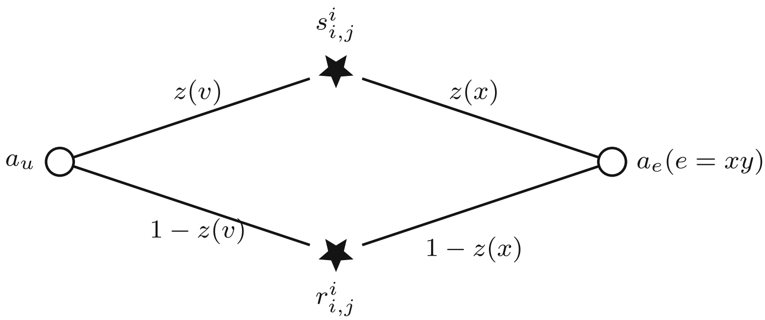

Fig. 2 The expected coverage of the vertices $s^i_{i,j}$ and $r^i_{i,j}$ due to selection of $\tilde{v}$ for some $v \in V_i$ and $\tilde{v}_e$ for some $e = xy \in E_{i,j}$

Proof Note that $w(s^i_{i,j}) = w(r^i_{i,j}) = -1$ and the probabilities of the edges $a_u s^i_{i,j}$, $a_u r^i_{i,j}$, $a_{xy} s^i_{i,j}$ and $a_{xy} r^i_{i,j}$ are $z(v)$, $1 - z(v)$, $z(x)$ and $1 - z(x)$, respectively. See Fig. 2 for more clarity. Then, we have $\mathcal{C}(s^i_{i,j}, \{a_u, a_{xy}\}) = -1 \times (\max(z(v), z(x))) = -(\max(z(v), z(x)))$, and $\mathcal{C}(r^i_{i,j}, \{a_u, a_{xy}\}) = -(\max(1 - z(v), 1 - z(x)))$. □

To maximize the expected coverage, we need the coverage of $r^i_{i,j}$ and $s^i_{i,j}$ by the pair of vertices a_u and a_{xy} is as maximum as possible. This implies the following corollary from Lemma 5.

Corollary 1 *The expected coverage of vertices* $\{s^i_{i,j}, r^i_{i,j}\}$ *by* $\{a_u, a_{xy}\}$ *is maximum when* $u = x$. *In this case,* $\mathcal{C}(\{s^i_{i,j}, r^i_{i,j}\}, \{a_u, a_{xy}\}) = -1$.

3.3 Equivalence

Now we show the equivalence of both the problems. More precisely, the graph G has a k-clique if and only if H has a k' sized vertex set that achieves the expected coverage of value at least t'.

Lemma 6 *If (G, k) is a YES-instance of the* MULTI-COLORED CLIQUE *problem, then (H, w, p, k', t') is also a YES-instance of the* EXPECTED COVERAGE *problem.*

Proof Let $K = \{u_1, u_2, \ldots, u_k\}$ be a k-clique in G such that for $i \in [k]$, $u_i \in V_i$. Now we construct a feasible solution $S \subseteq V(H)$ for the instance (H, w, p, k', t') of the EXPECTED COVERAGE problem. Let $S = \{a_{u_i} \mid i \in [k]\} \cup \{a_{u_i u_j} \mid 1 \leq i < j \leq k\}$. Note that the size of S is exactly the budget k'. For each $i \in [k]$, the set S is covering only the vertices t_i, q_i and a_{u_i} in H_i since $S \cap V(H_i) = \{a_{u_i}\}$. Similarly, for each $1 \leq i < j \leq k$, the set S is covering only the vertices t_i, q_i and a_{u_i} in H_i since $S \cap V(H_i) = \{a_{u_i}\}$. The expected coverage by the set S is given as:

$$\begin{aligned}\mathcal{C}(V(H), S) &= \sum_{i \in [k]} \Big(\mathcal{C}(\{t_i, a_{u_i}, q_i\}, a_{u_i})\Big) + \sum_{1 \leq i < j \leq k} \Big(\mathcal{C}(\{t_{i,j}, a_{u_i u_j}, q_{i,j}\}, a_{u_i u_j})\Big) \\ &\quad + \sum_{1 \leq i < j \leq k, \ell \in \{i,j\}} \Big(\mathcal{C}(s^{\ell}_{i,j}, \{a_{u_\ell}, a_{u_i u_j}\}) + \mathcal{C}(r^{\ell}_{i,j}, \{a_{u_\ell}, a_{u_i u_j}\})\Big) \\ &= \sum_{i \leq [k]} 2k^2 + \sum_{1 \leq i < j \leq k} 2k^2 + \sum_{1 \leq i < j \leq k} -2 = \left(k + \binom{k}{2}\right)(2k^2) - 2\binom{k}{2} \\ &= (k^2 + k)k^2 - (k^2 - k) = k^4 + k^3 - k^2 + k = t'.\end{aligned}$$

We apply the Corollary 1 in the second step to replace the exact value of expected coverage. Thus, we showed that $\mathcal{C}(V(H), S) = k^4 + k^3 - k^2 + k = t'$. Therefore, the set S is a feasible solution to the instance (H, w, p, k', t') of the EXPECTED COVERAGE problem. □

Now we prove the other direction of equivalence.

Lemma 7 *If (H, w, p, k', t') is a YES-instance of the* EXPECTED COVERAGE *problem then (G, k) is a YES-instance of the* MULTI-COLORED CLIQUE *problem.*

Proof Let S be a feasible solution to the instance (H, w, p, k', t') of the EXPECTED COVERAGE problem. The feasibility of S ensures that every gadget has a selection vertex from the set S. More specifically, each gadget contributes an expected coverage of $2k^2$. Then $\mathcal{C}(V(H) \setminus R, S) = k'(2k^2) = k^4 + k^3$ since S is a feasible solution.

There are $2\binom{k}{2}$ pairs of $s^i_{i,j}$ and $r^i_{i,j}$ connector vertices in H. By Lemma 5, each pair can contribute at most -1. Then, the value $k - k^2$ can be achieved only when each pair is contributing exactly -1. From Corollary 1, for each $1 \leq i < j \leq k$, the pair $r^i_{i,j}$ and $s^i_{i,j}$ together can contribute exactly -1 when $a_u \in S$ and $a_{uy} \in S$ for some $u \in V_i$ and $y \in V_j$. By construction of H, there is an edge between the vertices u and y in G. Let $K = \{u \in V_i \mid a_u \in S\}$ be a k sized vertex set from V. For every pair of distinct vertices in K will have en edge between them in G and thus form a k-clique in G. □

Thus, we state the following theorem using the Lemmas 2, 6 and 7.

Theorem 2 *The* EXPECTED COVERAGE *problem is W[1]-hard for the parameter pathwidth.*

Proof Given an instance (G, k) of the MULTI-COLORED CLIQUE problem, the instance (H, w, p, k', t') is constructed in polynomial time where $k' = k + \binom{k}{2}$ and $t' = k^4 + k^3 - k^2 + k$. From Lemma 2, we know that $\mathbf{pw}(H)$ is quadratic function of k. Finally, Lemmas 6 and 7 it follows that the instance (H, w, p, k', t') of the

EXPECTED COVERAGE problem output by the reduction is equivalent to the instance (G, k) of the MULTI-COLORED CLIQUE problem that was input to the reduction. Since the MULTI-COLORED CLIQUE problem is W[1]-hard for the parameter k, it follows that the EXPECTED COVERAGE problem is W[1]-hard for the parameter pathwidth of the input graph. □

Moreover, the parameterized reduction preserves the parameters k', t' and pathwidth of the constructed graph as a functions of k. That is, $k' = k + \binom{k}{2}$, $t' = k^4 + k^3 - k^2 + k$ and pathwidth of the graph H is $\mathcal{O}(k^2)$. Further, observe that the number of negative demand vertices in the reduced graph is $4\binom{k}{2} = \mathcal{O}(k^2)$. Thus, we conclude the section with following corollary.

Corollary 2 *The* EXPECTED COVERAGE *problem is W[1]-hard for any combination of following parameters, (i) budget k, (ii) number of negative demand vertices, (iii) pathwidth, (iv) vertex cover number, (v) feedback vertex set number and (vi) distance to star forest number of the input graph.*

4 FPT Algorithm for the EXPECTED COVERAGE problem Parameterized by Treewidth on Bounded Degree Graphs

While, as we have seen, the EXPECTED COVERAGE problem is W[1]-hard for the parameter pathwidth of the input graph, we complement the lower bound with an FPT algorithm for the combined treewidth and maximum vertex degree parameters.

Let (G, w, p, k) be an input to the optimization version of the EXPECTED COVERAGE problem. Let (T, X) be a nice tree decomposition of G with treewidth **tw**. Narayanaswamy et al. [33] introduced the notion of best neighbor to solve the EXPECTED COVERAGE problem with non-negative weights on bounded treewidth graphs. Consider a set $S \subseteq V$ of size k. In the LRO model, for each vertex u (with $u \in N[S]$), there exists a unique vertex in S called best neighbor of u in S, denoted by $bn(u, S)$ such that $\mathcal{C}(u, S) = \mathcal{C}(u, bn(u, S))$.

Definition 1 (Narayanaswamy et al. [33]) Given a vertex u and a set S with $u \in N[S]$, by $bn(u, S)$ we denote the *best neighbor* of u in S defined as follows:

$$bn(u, S) = \begin{cases} u & \text{if } u \in S, \\ v & \text{if} u \notin S \text{ then } v = \arg\max_{v' \in (N(u) \cap S)} p(uv'). \end{cases}$$

If $u \notin N[S]$ then $bn(u, S)$ is undefined. We use the fact that the graph G has bounded maximum degree. We define a structural ordering called *neighborhood indexing* on the neighborhood of each vertex. This LRO specific intuitions "*best neighbor*" and "*neighborhood indexing*", help us to solve the problem efficiently in tree decomposition.

Neighborhood indexing We define an ordered indexing on the neighborhood of each vertex $v \in V$. For each vertex $v \in V$, we order the vertices in $N(v)$ based on the survival probability of the edge connected to v in non-increasing order. Let $D_v = \{u_1, u_2, \ldots, u_{deg(v)}\}$ be the ordering of the vertices as described above. Let $\mathfrak{N}_v : [deg(v)] \to N(v)$ be a function on input a positive integer $r \leq deg(v)$ outputs the r^{th}-vertex u from the ordered set D_v.

4.1 Solution structure

For each node $\mathbf{i}$ in T, we compute two tables $\texttt{Sol}_\mathbf{i}$ and $\texttt{Val}_\mathbf{i}$. The rows of both tables are indexed by 3-tuple which we refer to as states. Let $\mathcal{S}_\mathbf{i}$ denote the set of all states associated with node $\mathbf{i}$. For a state $s \in \mathcal{S}_\mathbf{i}$, the DP formulation gives a recursive definition of the values $\texttt{Sol}_\mathbf{i}[s]$ and $\texttt{Val}_\mathbf{i}[s]$. $\texttt{Sol}_\mathbf{i}[s]$ is a set $S \subseteq X_\mathbf{i}{}^+$, an optimal solution for the instance specified by the state s. $\texttt{Val}_\mathbf{i}[s]$ is the value $\mathcal{C}(X_\mathbf{i}{}^+ \setminus \alpha^{-1}(0), \texttt{Sol}_\mathbf{i}[s])$ where α is an element of the state s defined below. A state $s \in \mathcal{S}_\mathbf{i}$ is a tuple (ℓ, α, β), where

- $0 \leq \ell \leq k$ is an integer and specify the size of $\texttt{Sol}_\mathbf{i}[s]$,
- $\alpha : X_\mathbf{i} \to \{0, 1\}$ is an indicator function for the vertices in $X_\mathbf{i}$. This specifies the constraint that whether a vertex in $X_\mathbf{i}$ is considered for the coverage.
- $\beta : X_\mathbf{i} \to \{-1, 0, 1, \ldots, \Delta\}$ is a function. This specifies the constraint that $X_\mathbf{i} \cap \texttt{Sol}_\mathbf{i}[s] = \beta^{-1}(0)$ and for each vertex $u \in X_\mathbf{i}$ with $\beta(u) \neq 0$, (i) if $\beta(u) = -1$ then $N(u) \cap \texttt{Sol}_\mathbf{i}[s] = \emptyset$, and (ii) if $\beta(u) > 0$ then $bn(u, \texttt{Sol}_\mathbf{i}[s]) = \mathfrak{N}_u(\beta(u))$.

An instance of the EXPECTED COVERAGE problem at the state s is $(G_\mathbf{i}, w_\mathbf{i}, p_\mathbf{i}, \ell)$ where the functions $w_\mathbf{i}$ and $p_\mathbf{i}$ are obtained from w and p for the domain $X_\mathbf{i}{}^+$ and $E(G_\mathbf{i})$, respectively. Additionally, the solution should satisfy the constraints specified by the state s. A state s is said to be *invalid* if there is no feasible solution that satisfies the constraints specified by s. We define one more notion of validity of the states called "locally valid". A state s is said to be *locally valid* if the following properties are satisfied.

- $\ell \geq |\beta^{-1}(0)|$, and
- for each vertex $u \in X_\mathbf{i}$, if $\beta(u) = -1$ then $N(u) \cap \beta^{-1}(0) = \emptyset$, and if $\beta(u) > 0$, then $\mathfrak{N}_u(\beta(u)) \in \big((X_\mathbf{i}{}^+ \setminus X_\mathbf{i}) \cup \beta^{-1}(0)\big)$.

If a state s is not locally valid, then s is an invalid state.

State induced at a node in T For any set $S \subseteq X_\mathbf{i}{}^+$ of size at most k and a function $\alpha : X_\mathbf{i} \to \{0, 1\}$ with $S \cap X_\mathbf{i} \subseteq \alpha^{-1}(1)$, we say that the pair (S, α) *induces* a state $s = (\ell, \alpha, \beta)$ at the node $\mathbf{i}$, and s is defined as follows:

- $\ell = |D|$.
- $\beta : X_\mathbf{i} \to \{-1, 0, 1, \ldots, \Delta\}$ is defined as follows: for each $u \in X_\mathbf{i} \cap D$, $\beta(u) = 0$, for each $u \in X_\mathbf{i} \cap N(D)$, $\beta(u) = \mathfrak{N}_v^{-1}(bn(u, D))$, and for each $u \in X_\mathbf{i} \setminus N[D]$, $\beta(u) = -1$.

4.2 Recursive definition of $\texttt{Sol}_{\mathbf{i}}$ and $\texttt{Val}_{\mathbf{i}}$

For each node $\mathbf{i}$ in T and a locally valid state $s = (\ell, \alpha, \beta) \in \mathcal{S}_{\mathbf{i}}$, we show how to compute $\texttt{Sol}_{\mathbf{i}}[s]$ and $\texttt{Val}_{\mathbf{i}}[s]$ from the tables at the children of $\mathbf{i}$. $\texttt{Sol}_{\mathbf{i}}[s]$ and $\texttt{Val}_{\mathbf{i}}[s]$ are recursively defined below and we prove a statement on the structure of an optimal solution based on the type of the node $\mathbf{i}$ in T. These statements are used in Section 4.3 to prove the correctness of the bottom-up evaluation.

Leaf Node Let $\mathbf{i}$ be a leaf node with bag $X_{\mathbf{i}} = \emptyset$. The state set $\mathcal{S}_{\mathbf{i}}$ is a singleton set with a state $s = (0, \emptyset \to \{0, 1\}, \emptyset \to \{-1, 0, 1, \ldots, \Delta\})$. Therefore, $\texttt{Sol}_{\mathbf{i}}[s] = \emptyset$ and $\texttt{Val}_{\mathbf{i}}[s] = 0$. This can be computed in constant time.

Lemma 8 *The table entries for the state s at a leaf node are computed optimally.*

Proof The correctness follows from the fact that the graph $G_{\mathbf{i}}$ is a null graph. Thus, for a null graph a valid state s, empty set with coverage value zero is the only optimal solution. □

Introduce Node Let $\mathbf{i}$ be an introduce node with child $\mathbf{j}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} \cup \{v\}$ for some $v \notin X_{\mathbf{j}}$. Since $\mathbf{i}$ is an introduce node, all the neighbors of v in $G_{\mathbf{i}}$ are in $X_{\mathbf{i}}$. That is, $N(v) \cap {X_{\mathbf{i}}}^+ \subseteq X_{\mathbf{i}}$. We define the state $s_{\mathbf{j}}$, and define $\texttt{Sol}_{\mathbf{i}}[s]$ in terms of $\texttt{Sol}_{\mathbf{j}}[s_{\mathbf{j}}]$. To define the state $s_{\mathbf{j}}$, we consider two cases depend on the value of $\beta(v)$.

Case $\boldsymbol{\beta(v) \neq 0}$ In this case, the desired solution for the state s must not contain the vertex v. We define the functions $\alpha_{\mathbf{j}} : X_{\mathbf{j}} \to \{0, 1\}$ and $\beta_{\mathbf{j}} \to \{-1, 0, 1, \ldots, \Delta\}$ from α and β by excluding the vertex v in the domains of both functions, respectively. We define $s_{\mathbf{j}} = (\ell, \alpha_{\mathbf{j}}, \beta_{\mathbf{j}})$. If the state $s_{\mathbf{j}}$ at the node $\mathbf{j}$ is invalid, then the state s at the node $\mathbf{i}$ is invalid. Therefore, we consider that the sate $s_{\mathbf{j}}$ is valid. Then, the solution for the state s is defined as follows:

$$\texttt{Sol}_{\mathbf{i}}[s] = \texttt{Sol}_{\mathbf{j}}[s_{\mathbf{j}}], \tag{1}$$

and

$$\texttt{Val}_{\mathbf{i}}[s] = \begin{cases} \texttt{Val}_{\mathbf{j}}[s_{\mathbf{j}}] + \mathcal{C}(v, \beta^{-1}(0)) & \text{if } \alpha(v) = 1, \\ \texttt{Val}_{\mathbf{j}}[s_{\mathbf{j}}] & \text{otherwise.} \end{cases} \tag{2}$$

Case $\boldsymbol{\beta(v) = 0}$ In this case, the desired solution for the state s must contain the vertex v. We have $\alpha(v) = 1$ since s is locally valid. The key idea is to figure out the vertices those will have v as its best neighbor in the desired solution. Then, find a suitable optimal state at node $\mathbf{j}$ and complete the bottom-up computation. Next we find set of all vertices that will have v as its best neighbor in the desired solution. Let

$D_v = \{u \in X_\mathbf{i} \mid \beta(u) > 0 \text{ and } \mathfrak{N}_u(\beta(u)) = v\}$. We enumerate all possible subsets of D_v to find such a set. For each $D \subseteq D_v$, we define the following:

- $\mathcal{F}_D = \{f : D \to \{-1, 0, 1, \ldots, \Delta\} \mid \forall v \in D, f(v) \neq -1 \Rightarrow f(v) > \beta(v)\}$
- $\alpha_\mathbf{j}^D : X_\mathbf{j} \to \{0, 1\}$ such that for each $u \in X_\mathbf{j}$ $\alpha_\mathbf{j}^D(u) = \begin{cases} 0 & \text{if } u \in D, \\ \alpha(u) & \text{otherwise.} \end{cases}$

For a set $D \subseteq D_v$ and a function $f \in \mathcal{F}_D$, we define the following:

- $\beta_\mathbf{j}^{D,f} : X_\mathbf{j} \to \{-1, 0, 1, \ldots, \Delta\}$ such that for each $u \in X_\mathbf{j}$,

$$\beta_\mathbf{j}^{D,f}(u) = \begin{cases} f(u) & \text{if } u \in D, \\ \beta(u) & \text{otherwise.} \end{cases}$$

- $s_\mathbf{j}^{D,f} = (\ell - 1, \alpha_\mathbf{j}^D, \beta_\mathbf{j}^{D,f})$.

An optimal set $D \subset D_v$ and $f \in \mathcal{F}_D$ can be computed as follows:

$$D, f = \underset{D' \subseteq D_v, f' \in \mathcal{F}'_D \mid s_\mathbf{j}^{D',f'} \text{ is valid}}{\arg\max} \mathtt{Val}_\mathbf{j}[s_\mathbf{j}^{D',f'}] + \mathcal{C}(D' \cup \{v\}, v). \tag{3}$$

If no such valid state found in $\mathbf{j}$, then we mark s in $\mathbf{i}$ is also invalid.

Define $s_\mathbf{j} = s_\mathbf{j}^{D,f} = (\ell - 1, \alpha_\mathbf{j}^D, \beta_\mathbf{j}^{D,f})$. Then, the solution for the state s is defined as follows:

$$\mathtt{Sol}_\mathbf{i}[s] = \mathtt{Sol}_\mathbf{j}[s_\mathbf{j}] \cup \{v\} \tag{4}$$

and

$$\mathtt{Val}_\mathbf{i}[s] = \mathtt{Val}_\mathbf{j}[s_\mathbf{j}] + \mathcal{C}(D, v) + w(v) \tag{5}$$

The time to compute the state $s_\mathbf{j}$ is depending on the size of $\mathcal{F}$. Therefore, $s_\mathbf{j}$ can be computed in $\mathcal{O}^*(\Delta^{\mathbf{tw}})$ time.

Lemma 9 *Let **i** be an introduce node in* T*, and let s and s_j be as defined above. If S is an optimal solution for the state s, then there exists a state $\hat{s}$ at the node **j** such that the set $S \setminus \{v\}$ is an optimal solution for the state $\hat{s}$, and $\mathtt{Val}_j[s_j] = \mathtt{Val}_j[\hat{s}]$.*

Proof Since S is an optimal solution to the state s, the pair (S, α) induces the state s at the node $\mathbf{i}$. We consider two cases based on whether $v \in S$ or not. First consider the case $v \in S$. In this case, observe that $\beta(v) = 0$. Consider the set $D = \{u \in X_\mathbf{i} \mid bn(u, S) = v\}$ and the function $\alpha_\mathbf{j}^D$ as defined above for the case $\beta(v) = 0$. Since $(S \setminus \{v\}) \cap X_\mathbf{j} \subseteq \alpha_\mathbf{j}^{D^{-1}}(1)$ by definition of $\alpha_\mathbf{j}^D$, let $\hat{s} = (\ell - 1, \alpha_\mathbf{j}^D, \hat{\beta}_\mathbf{j})$ be the state induced by the pair $(S \setminus \{v\}, \alpha_\mathbf{j}^D)$ at the node $\mathbf{j}$. Let us define the function $f : D \to \{-1, 0, 1, \ldots, \Delta\}$ such that for each $u \in D$, $f(u) = \hat{\beta}_\mathbf{j}(u)$. Note that the state $\hat{s}$ and $s_\mathbf{j}^{D,f}$ are same. Observe that from (3) and optimality of S for the state s,

$\mathtt{Val}_{\mathbf{j}}[\hat{s}] = \mathtt{Val}_{\mathbf{j}}[s_{\mathbf{j}}]$. Then, the expected coverage $\mathcal{C}_{\mathbf{i}}({X_{\mathbf{i}}}^+ \setminus \alpha^{-1}(0), S)$ can be written as follows:

$$\begin{aligned}
\mathcal{C}({X_{\mathbf{i}}}^+ \setminus \alpha^{-1}(0), S) &= \mathcal{C}({X_{\mathbf{i}}}^+ \setminus (\alpha^{-1}(0) \cup D \cup v), S) + \mathcal{C}(D_v, S) + \mathcal{C}(v, S) \\
&= \mathcal{C}({X_{\mathbf{i}}}^+ \setminus (\alpha^{-1}(0) \cup D \cup v), S \setminus \{v\}) + \mathcal{C}(D, v) + w(v) \\
&= \mathcal{C}(X_{\mathbf{j}}^+ \setminus (\alpha^{-1}(0) \cup D), S \setminus \{v\}) + \mathcal{C}(D, v) + w(v) \\
&= \mathcal{C}(X_{\mathbf{j}}^+ \setminus \alpha_{\mathbf{j}}^{-1}(0), S \setminus \{v\}) + \mathcal{C}(D, v) + w(v) \\
&= \mathtt{Val}_{\mathbf{j}}[\hat{s}] + \mathcal{C}(D, v) + w(v) \\
&= \mathtt{Val}_{\mathbf{j}}[s_{\mathbf{j}}] + \mathcal{C}(D, v) + w(v).
\end{aligned}$$

Note that the term $\mathcal{C}(D, v) + w(v)$ is independent of the set S. Thus, the set $S \setminus \{v\}$ is an optimal solution for the state $\hat{s}$ at the node $\mathbf{j}$.

Next we consider the case $v \notin S$. In this case, observe that $\beta(v) \neq 0$ and $S \setminus \{v\} = S$. Consider the $\alpha_{\mathbf{j}}$ as defined above for the case $\beta(v) \neq 0$. Let $\hat{s} = (\ell, \alpha_{\mathbf{j}}, \hat{\beta}_{\mathbf{j}}) \in \mathcal{S}_{\mathbf{j}}$ be the state induced by the set S at the node $\mathbf{j}$. Observe that the functions $\beta_{\mathbf{j}}$ and $\hat{\beta}_{\mathbf{j}}$ are same. Thus, $\hat{s}$ and $s_{\mathbf{j}}$ are same. Then, the expected coverage $\mathcal{C}({X_{\mathbf{i}}}^+ \setminus \alpha^{-1}(0), S)$ can be written as follows:

$$\mathcal{C}({X_{\mathbf{i}}}^+ \setminus \alpha^{-1}(0), S) = \begin{cases} \mathtt{Val}_{\mathbf{j}}[\hat{s}] + \mathcal{C}(v, S) & \text{if } \alpha(v) = 1, \\ \mathtt{Val}_{\mathbf{j}}[\hat{s}] & \text{if } \alpha(v) = 0. \end{cases}$$

Thus, the set S is an optimal optimal solution for the state $\hat{s}$ at the node $\mathbf{j}$. □

Forget Node Let $\mathbf{i}$ be a forget node with child $\mathbf{j}$ such that $X_{\mathbf{i}} = X_{\mathbf{j}} \setminus \{v\}$ for some $v \in X_{\mathbf{j}}$. Since $\mathbf{i}$ is a forget node, $N(v) \subseteq {X_{\mathbf{i}}}^+$. We define the state $s_{\mathbf{j}} = (\ell, \alpha_{\mathbf{j}}, \beta_{\mathbf{j}})$ and define $\mathtt{Sol}_{\mathbf{i}}[s]$ in terms of $\mathtt{Sol}_{\mathbf{j}}[s_{\mathbf{j}}]$. The state s does not impose any constraint on v since $v \notin X_{\mathbf{i}}$. So, we try all possible values of α and β for v, and find an optimal one. We define $\alpha_{\mathbf{j}} : X_{\mathbf{j}} \to \{0, 1\}$ such that for each $u \in X_{\mathbf{j}}$,

$$\alpha_{\mathbf{j}}[u] = \begin{cases} 1 & \text{if } u = v, \\ \alpha(u) & \text{otherwise.} \end{cases}$$

Therefore, we consider all possible values of $\beta_{\mathbf{j}}(v)$ to define the state $s_{\mathbf{j}}$. For each $z \in \{0, 1, \ldots, deg(v)\}$, we define ${\beta_{\mathbf{j}}}^z : X_{\mathbf{j}} \to \{-1, 0, 1, \ldots, \Delta\}$ such that for each $u \in X_{\mathbf{j}}$,

$$\beta_{\mathbf{j}}^z(u) = \begin{cases} z & \text{if } u = v, \\ \beta(u) & \text{otherwise.} \end{cases}$$

For each $z \in \{0, 1, \ldots, deg(v)$, we define $s_{\mathbf{j}}^z = (\ell, \alpha_{\mathbf{j}}, \beta_{\mathbf{j}}^z)$. If for each $z \in \{0, 1, \ldots, deg(v)\}$, the state $s_{\mathbf{j}}^z$ at the node $\mathbf{j}$ is invalid then the state s at the node $\mathbf{i}$ is invalid. Therefore, we consider that there exists a $z \in \{0, 1, \ldots, deg(v)\}$ such that the state $s_{\mathbf{j}}^z$ at the node $\mathbf{j}$ is valid. Let

$$z' = \arg \max_{z \in \{0,1,\ldots,deg(v)\} | s_{\mathbf{j}}^z \text{ is valid}} \mathtt{Val}_{\mathbf{j}}[s_{\mathbf{j}}^z]. \tag{6}$$

Define $s_{\mathbf{j}} = s_{\mathbf{j}}^{z'} = (\ell, \alpha_{\mathbf{j}}, \beta_{\mathbf{j}}^{z'})$. Then, the solution for the state s is defined as follows:

$$\texttt{Sol}_{\mathbf{i}}[s] = \texttt{Sol}_{\mathbf{j}}[s_{\mathbf{j}}], \tag{7}$$

and

$$\texttt{Val}_{\mathbf{i}}[s] = \texttt{Val}_{\mathbf{j}}[s_{\mathbf{j}}]. \tag{8}$$

The state $s_{\mathbf{j}}$ can be computed in $\mathcal{O}(\Delta)$ time.

Lemma 10 *Let **i** be a forget node in* T*, and let s and s_j be as defined above. If S is an optimal solution for the state s, then there exists a state $\hat{s}$ at the node **j** such that the set S is an optimal solution for the state $\hat{s}$, and $\texttt{Val}_j[s_j] = \texttt{Val}_j[\hat{s}]$.*

Proof Since S is an optimal solution for the state s, the pair (S, α) induces the state s at the node **i**. Consider the function $\alpha_{\mathbf{j}}$ as defined above. Let $\hat{s} = (\ell, \alpha_{\mathbf{j}}, \hat{\beta}_{\mathbf{j}})$ be the state induced by the pair $(S, \alpha_{\mathbf{j}})$ at the node **j**. Let $z = \hat{\beta}_{\mathbf{j}}(v)$. Note that the functions $\beta_{\mathbf{j}}^{z}$ and $\hat{\beta}_{\mathbf{j}}$ are same, and thus the states $\hat{s}$ and $s_{\mathbf{j}}^{z}$ are same. From (6) and the optimality of the set S for the state s, $\texttt{Val}_{\mathbf{j}}[\hat{s}] = \texttt{Val}_{\mathbf{j}}[s_{\mathbf{j}}]$. Then, the expected coverage $\mathcal{C}(X_{\mathbf{i}}{}^{+} \setminus \alpha^{-1}(0), S)$ can be written as follows:

$$\mathcal{C}(X_{\mathbf{i}}{}^{+} \setminus \alpha^{-1}(0), S) = \mathcal{C}(X_{\mathbf{j}}^{+} \setminus \alpha_{\mathbf{j}}^{-1}(0), S) = \texttt{Val}_{\mathbf{j}}[\hat{s}] = \texttt{Val}_{\mathbf{j}}[s_{\mathbf{j}}].$$

Thus, the set S is an optimal solution for the state $\hat{s}$ at the node **j**. $\square$

Join Node Let **i** be a join node with children **j** and **h** such that $X_{\mathbf{i}} = X_{\mathbf{j}} = X_{\mathbf{h}}$. We define two states $s_{\mathbf{j}} = (\ell_{\mathbf{j}}, \alpha_{\mathbf{j}}, \beta_{\mathbf{j}})$ and $s_{\mathbf{h}} = (\ell_{\mathbf{h}}, \alpha_{\mathbf{h}}, \beta_{\mathbf{h}})$ at nodes **j** and **h**, respectively, and define $\texttt{Sol}_{\mathbf{i}}[s]$ in terms of $\texttt{Sol}_{\mathbf{j}}[s_{\mathbf{j}}]$ and $\texttt{Sol}_{\mathbf{h}}[s_{\mathbf{h}}]$. Among ℓ vertices in the solution to be computed, $|\beta^{-1}(0)|$ many vertices are taken from $X_{\mathbf{i}}$ and $\ell - |\beta^{-1}(0)|$ many vertices will be chosen from $X_{\mathbf{i}}{}^{+} \setminus X_{\mathbf{i}}$. Since $X_{\mathbf{i}}{}^{+} \setminus X_{\mathbf{i}}$ is a disjoint union of the sets $X_{\mathbf{j}}^{+} \setminus X_{\mathbf{j}}$ and $X_{\mathbf{h}}^{+} \setminus X_{\mathbf{h}}$, we consider a parameter z to partition the value $\ell - |\beta^{-1}(0)|$. For each $0 \le z \le \ell - |\beta^{-1}(0)|$, let $\ell_{\mathbf{j}}^{z} = |\beta^{-1}(0)| + z$ and $\ell_{\mathbf{h}}^{z} = b - z$: we consider the states at nodes **j** and **h** with budget $\ell_{\mathbf{j}}^{z}$ and $\ell_{\mathbf{h}}^{z}$, respectively. Let

$$D = \{u \in X_{\mathbf{i}} \mid \beta(u) \notin \{-1, 0\} \text{ and } \mathfrak{N}_u(\beta(u)) \in X_{\mathbf{i}}{}^{+} \setminus X_{\mathbf{i}}\}.$$

For each $u \in D$, $\mathfrak{N}_u(\beta(u))$ is in either $X_{\mathbf{j}}^{+} \setminus X_{\mathbf{j}}$ and $X_{\mathbf{h}}{}^{+} \setminus X_{\mathbf{h}}$. Let $\mathcal{D}_{\mathbf{j}} = \{u \in D \mid \mathfrak{N}_u(\beta(u)) \in X_{\mathbf{j}}{}^{+} \setminus X_{\mathbf{j}}\}$ and $\mathcal{D}_{\mathbf{h}} = \{u \in D \mid \mathfrak{N}_u(\beta(u)) \in X_{\mathbf{h}}{}^{+} \setminus X_{\mathbf{h}}\}$. Note that the set D is partitioned into $D_{\mathbf{j}}$ and $D_{\mathbf{h}}$. We define $\alpha_{\mathbf{j}} : X_{\mathbf{j}} \to \{0, 1\}$ such that for each $u \in X_{\mathbf{j}}$,

$$\alpha_{\mathbf{j}}(u) = \begin{cases} 0 & \text{if } u \in D_{\mathbf{h}} \text{ and } \alpha(u) = 1, \\ \alpha(u) & \text{otherwise.} \end{cases}$$

Similarly, we define $\alpha_{\mathbf{h}} : X_{\mathbf{h}} \to \{0, 1\}$ such that for each $u \in X_{\mathbf{h}}$,

$$\alpha_{\mathbf{h}}(u) = \begin{cases} 0 & \text{if } u \in D_{\mathbf{j}} \text{ and } \alpha(u) = 1, \\ \alpha(u) & \text{otherwise.} \end{cases}$$

Let $\mathcal{A} = \{a : D_{\mathbf{h}} \to \{-1, 0, 1, \ldots, \Delta\} \mid \forall u \in D_{\mathbf{h}},$ if $a(u) \neq -1$ then $\beta(u)+1 \leq a(u) \leq deg(u)\}$. For each $a \in \mathcal{A}$, let $\beta_{\mathbf{j}}^{a} : X_{\mathbf{j}} \to \{-1, 0, 1, \ldots, \Delta\}$ such that for each $u \in X_{\mathbf{j}}$,

$$\beta_{\mathbf{j}}^{a}(u) = \begin{cases} a(u) & \text{if } u \in D_{\mathbf{h}}, \\ \beta(u) & \text{otherwise.} \end{cases}$$

Let $\mathcal{B} = \{b : D_{\mathbf{j}} \to \{-1, 0, 1, \ldots, \Delta\} \mid \forall u \in D_{\mathbf{j}},$ if $b(u) \neq -1$ then $\beta(u) + 1 \leq b(u) \leq deg(u)\}$. For each $b \in \mathcal{B}$, let $\beta_{\mathbf{h}}^{b} : X_{\mathbf{j}} \to \{-1, 0, 1, \ldots, \Delta\}$ such that for each $u \in X_{\mathbf{j}}$,

$$\beta_{\mathbf{h}}^{b}(u) = \begin{cases} b(u) & \text{if } u \in D_{\mathbf{j}}, \\ \beta(u) & \text{otherwise.} \end{cases}$$

For each $0 \leq z \leq \ell - |\beta^{-1}(0)|$, $a \in \mathcal{A}$ and $b \in \mathcal{B}$, let $s_{\mathbf{j}}^{z,a} = ({\ell_{\mathbf{j}}}^{z}, \alpha_{\mathbf{j}}, {\beta_{\mathbf{j}}}^{a})$ and ${s_{\mathbf{h}}}^{z,b} = ({\ell_{\mathbf{h}}}^{z}, \alpha_{\mathbf{h}}, {\beta_{\mathbf{h}}}^{b})$. If for each $0 \leq z \leq \ell - |\beta^{-1}(0)|$, $a \in \mathcal{A}$ and $b \in \mathcal{B}$, either $s_{\mathbf{j}}^{z,a}$ or ${s_{\mathbf{h}}}^{z,b}$ is invalid, then s is invalid. Therefore, we consider that there exists a $0 \leq z \leq \ell - |\beta^{-1}(0)|$, $a \in \mathcal{A}$ and $b \in \mathcal{B}$ such that both $s_{\mathbf{j}}^{z,a}$ and ${s_{\mathbf{h}}}^{z,b}$ are valid. Further, we define the following tuple:

$$z', a', b' = \arg \max_{\substack{0 \leq z \leq \ell - |\beta^{-1}(0)|, \\ a \in \mathcal{A}, b \in \mathcal{B} | s_{\mathbf{j}}^{z,a} \text{ and } s_{\mathbf{h}}^{z,b} \text{ are valid}}} \mathtt{Sol}_{\mathbf{j}}[s_{\mathbf{j}}^{z,a}] + \mathtt{Sol}_{\mathbf{h}}[s_{\mathbf{h}}^{z,b}]. \tag{9}$$

Define $s_{\mathbf{j}} = s_{\mathbf{j}}^{z',a'} = (\ell_{\mathbf{j}}^{z'}, \alpha_{\mathbf{j}}, \beta_{\mathbf{j}}^{a'})$ and $s_{\mathbf{h}} = s_{\mathbf{h}}^{z',b'} = (\ell_{\mathbf{h}}^{z'}, \alpha_{\mathbf{h}}, \beta_{\mathbf{h}}^{b'})$. Then, the solution for the state s is defined as follows:

$$\mathtt{Sol}_{\mathbf{i}}[s] = \mathtt{Sol}_{\mathbf{j}}[s_{\mathbf{j}}] \cup \mathtt{Sol}_{\mathbf{h}}[s_{\mathbf{h}}], \tag{10}$$

and

$$\mathtt{Val}_{\mathbf{i}}[s] = \mathtt{Val}_{\mathbf{j}}[s_{\mathbf{j}}] + \mathtt{Val}_{\mathbf{h}}[s_{\mathbf{h}}] - \mathcal{C}(\alpha^{-1}(1) \setminus D, \beta^{-1}(0)). \tag{11}$$

The subtracted term in the (11) is the over-counting term of the combined solution that is obtained from the states $s_{\mathbf{j}}$ and $s_{\mathbf{h}}$. The time to compute the states $s_{\mathbf{j}}$ and $s_{\mathbf{h}}$ is depending on the sizes of the functions $\mathcal{A}$ and $\mathcal{B}$. Therefore, the states $s_{\mathbf{j}}$ and $s_{\mathbf{h}}$ can be computed in $|\mathcal{A}| \cdot |\mathcal{B}| = \mathcal{O}(\Delta^{\mathbf{tw}})$ time.

Lemma 11 *Let* $\boldsymbol{i}$ *be a join node in* T*, and let* s*,* $s_{\boldsymbol{j}}$ *and* $s_{\boldsymbol{h}}$ *be as defined above. Let* S *be an optimal solution for the state* s*. Let* $S_{\boldsymbol{j}} = S \cap X_{\boldsymbol{j}}^{+}$ *and* $S_{\boldsymbol{h}} = S \cap X_{\boldsymbol{h}}^{+}$*. Then, there exists two states* $\hat{s}$ *and* $\tilde{s}$ *at nodes* $\boldsymbol{j}$ *and* $\boldsymbol{h}$*, respectively, such that the sets* $S_{\boldsymbol{j}}$ *and* $S_{\boldsymbol{h}}$ *are optimal solutions for the states* $\hat{s}$ *and* $\tilde{s}$*, respectively. Further,* $\mathtt{Val}_{\boldsymbol{j}}[s_{\boldsymbol{j}}] = \mathtt{Val}_{\boldsymbol{j}}[\hat{s}]$ *and* $\mathtt{Val}_{\boldsymbol{h}}[s_{\boldsymbol{h}}] = \mathtt{Val}_{\boldsymbol{h}}[\tilde{s}]$.

Proof Since S is an optimal solution for the state s, the pair (S, α) induces the state s at the node $\mathbf{i}$. Consider the set $D = D_{\mathbf{j}} \cup D_{\mathbf{h}}$, and the functions $\alpha_{\mathbf{j}}$ and $\alpha_{\mathbf{h}}$ as defined above. Let $\hat{s} = (\hat{\ell_{\mathbf{j}}}, \hat{\alpha_{\mathbf{j}}}, \hat{\beta_{\mathbf{j}}})$ be the state at the node $\mathbf{j}$ induced by the pair $(S_{\mathbf{j}}, \alpha_{\mathbf{j}})$. Let $\tilde{s} = (\tilde{\ell_{\mathbf{h}}}, \tilde{\alpha_{\mathbf{h}}}, \tilde{\beta_{\mathbf{h}}})$ be the state at the node $\mathbf{h}$ induced by the pair $(S_{\mathbf{h}}, \alpha_{\mathbf{h}})$. Let $z = \hat{\ell_{\mathbf{j}}} - |\beta^{-1}(0)|$. Define $a : D_{\mathbf{h}} \to \{-1, 0, 1, \ldots, \Delta\}$ such that for each $u \in \mathcal{D}_{\mathbf{h}}$, $a(u) = \hat{\beta}_{\mathbf{j}}(u)$. Then, define $b : D_{\mathbf{j}} \to \{-1, 0, 1, \ldots, \Delta\}$ such that for each $u \in \mathcal{D}_{\mathbf{j}}$, $a(u) = \tilde{\beta}_{\mathbf{h}}(u)$. Note that the functions $\hat{\beta}_{\mathbf{j}}$ and $\tilde{\beta}_{\mathbf{h}}$ are same as $\beta^a_{\mathbf{j}}$ and $\beta^b_{\mathbf{h}}$, respectively. Thus, the states $\hat{s}$ and $\tilde{s}$ are same as $s^{z,a}_{\mathbf{j}}$ and $s^{z,b}_{\mathbf{h}}$, respectively. From (9) and the optimality of the set S for the state s, $\texttt{Val}_{\mathbf{j}}[s_{\mathbf{j}}] = \texttt{Val}_{\mathbf{j}}[s_{\mathbf{j}}]$ and $\texttt{Val}_{\mathbf{h}}[s_{\mathbf{h}}] = \texttt{Val}_{\mathbf{h}}[s_{\mathbf{h}}]$. Then, the expected coverage $\mathcal{C}(X_{\mathbf{i}}{}^+ \setminus \alpha^{-1}(0))$ can be written as follows:

$$
\begin{aligned}
\mathcal{C}(X_{\mathbf{i}}{}^+ \setminus \alpha^{-1}(0), S) &= \mathcal{C}(X_{\mathbf{i}}{}^+ \setminus X_{\mathbf{i}}, S) + \mathcal{C}_{\mathbf{i}}(X_{\mathbf{i}} \setminus \alpha^{-1}(0), S) \\
&= \mathcal{C}(X^+_{\mathbf{j}} \setminus X_{\mathbf{j}}, S) + \mathcal{C}(X^+_{\mathbf{h}} \setminus X_{\mathbf{h}}, S) + \mathcal{C}(\alpha^{-1}(1) \setminus D, S) \\
&\quad + \mathcal{C}(D, S) \\
&= \mathcal{C}(X^+_{\mathbf{j}} \setminus X_{\mathbf{j}}, S_{\mathbf{j}}) + \mathcal{C}(X^+_{\mathbf{h}} \setminus X_{\mathbf{h}}, S_{\mathbf{h}}) + \mathcal{C}(D_{\mathbf{j}}, S) + \mathcal{C}(D_{\mathbf{h}}, S) \\
&\quad + \mathcal{C}(\alpha^{-1}(1) \setminus D, S) + \mathcal{C}(\alpha^{-1}(1) \setminus D, S) - \mathcal{C}(\alpha^{-1}(1) \setminus D, S) \\
&= \mathcal{C}(X^+_{\mathbf{j}} \setminus X_{\mathbf{j}}, S_{\mathbf{j}}) + \mathcal{C}(X^+_{\mathbf{h}} \setminus X_{\mathbf{h}}, S_{\mathbf{h}}) + \mathcal{C}(\alpha^{-1}(1) \setminus D_{\mathbf{h}}, S) \\
&\quad + \mathcal{C}(\alpha^{-1}(1) \setminus D_{\mathbf{j}}, S) - \mathcal{C}(\alpha^{-1}(1) \setminus D, S) \\
&= \mathcal{C}(X^+_{\mathbf{j}} \setminus X_{\mathbf{j}}, S_{\mathbf{j}}) + \mathcal{C}(X^+_{\mathbf{h}} \setminus X_{\mathbf{h}}, S_{\mathbf{h}}) + \mathcal{C}(\alpha^{-1}(1) \setminus D_{\mathbf{h}}, S_{\mathbf{j}}) \\
&\quad + \mathcal{C}(\alpha^{-1}(1) \setminus D_{\mathbf{j}}, S_{\mathbf{h}}) - \mathcal{C}(\alpha^{-1}(1) \setminus D, S) \\
&= \mathcal{C}(X^+_{\mathbf{j}} \setminus (\alpha^{-1}(0) \cup D_{\mathbf{h}}), S_{\mathbf{j}}) + \mathcal{C}(X^+_{\mathbf{h}} \setminus (\alpha^{-1}(0) \cup D_{\mathbf{j}}), S_{\mathbf{h}}) \\
&\quad - \mathcal{C}(\alpha^{-1}(1) \setminus D, \beta^{-1}(0)) \\
&= \mathcal{C}(X^+_{\mathbf{j}} \setminus \alpha^{-1}_{\mathbf{j}}(0), S_{\mathbf{j}}) + \mathcal{C}(X^+_{\mathbf{h}} \setminus \alpha^{-1}_{\mathbf{h}}(0), S_{\mathbf{h}}) \\
&\quad - \mathcal{C}(\alpha^{-1}(1) \setminus D, \beta^{-1}(0)) \\
&= \texttt{Val}_{\mathbf{j}}[\hat{s}] + \texttt{Val}_{\mathbf{h}}[\tilde{s}] - \mathcal{C}(\alpha^{-1}(1) \setminus D, \beta^{-1}(0)).
\end{aligned}
$$

If either $S_{\mathbf{j}}$ or $S_{\mathbf{h}}$ is not optimal to the state $\hat{s}$ or $\tilde{s}$, then it contradicts the optimality of the solution S for the state s. Thus, the sets $S_{\mathbf{j}}$ and $S_{\mathbf{h}}$ are optimal solutions for the states $\hat{s}$ and $\tilde{s}$, respectively. □

4.3 Bottom-up evaluation: Correctness of the DP formulation

Correctness Invariant For a node $\mathbf{i}$ and a valid state $s = (\ell, \alpha, \beta)$ at $\mathbf{i}$, the recursive definition in Section 4.2 ensures that

$$\texttt{Sol}_{\mathbf{i}}[s] = \underset{\substack{S \subseteq X_{\mathbf{i}}{}^+ \setminus \alpha^{-1}(0), |S| = \ell, \\ (S,\alpha) \text{ induces } s}}{\arg\max} \mathcal{C}(X_{\mathbf{i}}{}^+ \setminus \alpha^{-1}(0), S)$$

Lemma 12 *For each $\mathbf{i} \in V(\mathsf{T})$, and each valid state $s \in \mathcal{S}_{\mathbf{i}}$, the correctness invariant is maintained for $\texttt{Sol}_{\mathbf{i}}[s]$.*

Proof The proof is by induction on the height of a node in T. The height of a node $\mathbf{i}$ in a rooted tree T is the distance to the farthest leaf in the subtree rooted at $\mathbf{i}$. The base case is when $\mathbf{i}$ is a leaf node in T and height is zero, and the proof follows from Lemma 8. Let us assume that the claim is true for all nodes in T of height at most $h - 1 \geq 0$. We now prove that if the claim is true for all nodes of height at most

$h-1$, then it is true for a node of height h. Let $\mathbf{i}$ be a node of height $h \geq 1$. Since $\mathbf{i}$ is not a leaf node, its children are of height at most $h-1$. Therefore, by the induction hypothesis, the correctness invariant is maintained at all children of $\mathbf{i}$. We now prove that the correctness invariant is maintained at the node $\mathbf{i}$. Let S be an optimal solution for the state s. Then, we show that $\mathtt{Val}_{\mathbf{i}}[s] = \mathcal{C}(X_{\mathbf{i}} \setminus \alpha^{-1}(0), S)$. If $\mathbf{i}$ is an introduce node then from Lemma 9, we have that the set $S \setminus \{v\}$ is an optimal solution for the state $\hat{s} \in \mathcal{S}_{\mathbf{j}}$. Since $\mathbf{j}$ is at height at most $h-1$ and by induction hypothesis, the correctness invariant is maintained at state s of the node $\mathbf{i}$. If $\mathbf{i}$ is a forget node then from Lemma 10, we have that the set S is an optimal solution for the state $\hat{s} \in \mathcal{S}_{\mathbf{j}}$. Since $\mathbf{j}$ is at height at most $h-1$ and by induction hypothesis, the correctness invariant is maintained at state s of the node $\mathbf{i}$. If $\mathbf{i}$ is a join node then from Lemma 11, we have that the sets $S_{\mathbf{j}}$ and $S_{\mathbf{h}}$ (as defined in Lemma 11) are optimal solutions for the states $\hat{s} \in \mathcal{S}_{\mathbf{j}}$ and $\tilde{s} \in \mathcal{S}_{\mathbf{h}}$, respectively. Since $\mathbf{j}$ and $\mathbf{h}$ are at height at most $h-1$ and by induction hypothesis, the correctness invariant is maintained at state s of the node $\mathbf{i}$. This completes the proof. □

Thus we conclude the section with following theorem.

Theorem 3 *The* EXPECTED COVERAGE *problem can be solved optimally in time* $2^{\mathcal{O}(\boldsymbol{tw} \log \Delta)} n^{\mathcal{O}(1)}$.

Proof An optimal solution can be obtained the state $s = (k, \emptyset \to \{0,1\}, \emptyset \to \{-1, 0, 1, \ldots, \Delta\})$ at the node $\mathbf{r}$. That is, the set $\mathtt{Sol}_{\mathbf{r}}[s]$ is an optimal solution to the input instance of the EXPECTED COVERAGE problem. The correctness of the tables computation is proved in Lemma 12. Note that every node has a table of size $(k+1)(2\Delta+4)^{\mathbf{tw}}$ and each entry can be updated in time $\mathcal{O}(\Delta^{\mathbf{tw}})$. A nice tree decomposition with width $\mathbf{tw}$ and $\mathcal{O}(n \cdot \mathbf{tw})$ nodes can be computed in polynomial time [30] on inputting a tree decomposition with width $\mathbf{tw}$. Therefore, given a graph G and a tree decomposition of G with width $\mathbf{tw}$, the EXPECTED COVERAGE problem can be solved in time $2^{\mathcal{O}(\mathbf{tw} \log \Delta)} n^{\mathcal{O}(1)}$. □

5 Conclusion

In this article, we considered the EXPECTED COVERAGE problem. We focus on structural parameterization, due to the EXPECTED COVERAGE problem is W[2]-hard for the solution size k. In particular, we show the parameterized complexity of the EXPECTED COVERAGE problem with respect to the well-known graph parameters treewidth **tw**, pathwidth **pw**, vertex cover **vc**, feedback vertex set **fvs**, and distance to star forest **dsf**. Further, we observed from our reduction that the EXPECTED COVERAGE problem with respect to number of negative demand vertices as parameter is W[1]-hard. Finally, for the combined parameters treewidth and maximum degree ($\mathbf{tw} + \Delta$), we give an FPT algorithm to the EXPECTED COVERAGE problem.

Remaining open questions on the structurally parameterized complexity of the EXPECTED COVERAGE problem concern the tight fine-grained bounds. On the other hand, FPT approximation scheme for any of the above structural properties would be a good way to complement the parameterized hardness result of the Expected Coverage problem.

Funding Open access funding provided by University of Bergen (incl Haukeland University Hospital).

References

1. Ageev, A.A.: A criterion of polynomial-time solvability for the network location problem. In: Proceedings of the 2nd Integer Programming and Combinatorial Optimization Conference, Pittsburgh, PA, USA, May 1992, pp. 237–245 (1992)
2. Alber, J., Fellows, M.R., Niedermeier, R.: Polynomial-time data reduction for dominating set. J. ACM **51**(3), 363–384 (2004)
3. Alber, J., Fernau, H., Niedermeier, R.: Parameterized complexity: exponential speed-up for planar graph problems. J. Algorithms **52**(1), 26–56 (2004)
4. Berman, O., Drezner, Z., Krass, D.: Generalized coverage: New developments in covering location models. Comput. Oper. Res. **37**(10), 1675–1687 (2010)
5. Berman, O., Drezner, Z., Wesolowsky, G.O.: The maximal covering problem with some negative weights. Geographic. Anal. **41**(1), 30–42 (2009)
6. Bevern, R., Tsidulko, O.Y., Zschoche, P.: Fixed-Parameter Algorithms for Maximum-Profit Facility Location under Matroid Constraints. In: Algorithms and Complexity - 11Th International Conference, CIAC 2019, Rome, Italy, May 27-29, 2019, Proceedings, pp. 62–74 (2019)
7. Charikar, M., Guha, S.: Improved Combinatorial Algorithms for the Facility Location and K-Median Problems. In: 40Th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA, pp. 378–388 (1999)
8. Church, R., Velle, C.R.: The maximal covering location problem. Papers in Regional Science **32**(1), 101–118 (1974)
9. Colbourn, C.J., Xue, G.: A linear time algorithm for computing the most reliable source on a series-parallel graph with unreliable edges. Theor. Comput. Sci. **209**(1), 331–345 (1998)
10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized algorithms. Springer, Berlin (2015)
11. Demaine, E.D., Fomin, F.V., Hajiaghayi, M.T., Thilikos, D.M.: Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs. J. ACM **52**(6), 866–893 (2005)
12. Diestel, R.: Graph Theory, 4th Edition, Graduate texts in mathematics, vol. 173. Springer, Berlin (2012)
13. Ding, W.: Computing the Most Reliable Source on Stochastic Ring Networks. In: 2009 WRI World Congress on Software Engineering, vol. 1, pp. 345–347 (2009)
14. Ding, W.: Extended most reliable source on an unreliable general network. In: 2011 International Conference on Internet Computing and Information Services, pp. 529–533 (2011)
15. Ding, W., Xue, G.: A linear time algorithm for computing a most reliable source on a tree network with faulty nodes. Theoretical Computer Science **412**(3), 225–232 (2011). Combinatorial Optimization and Applications
16. Downey, R.G., Fellows, M.R.: Fixed Parameter Tractability and Completeness. In: Complexity Theory: Current Research, Dagstuhl Workshop, February 2-8, 1992, pp. 191–225 (1992)

17. Downey, R.G., Fellows, M.R.: Fundamentals of parameterized complexity. Texts in computer science. Springer, Berlin (2013)
18. Drange, P.G., Dregi, M.S., Fomin, F.V., Kreutzer, S., Lokshtanov, D., Pilipczuk, M., Pilipczuk, M., Reidl, F., Villaamil, F.S., Saurabh, S., Siebertz, S., Sikdar, S.: Kernelization and sparseness: the case of dominating set. In: Proceedings of the 33rd International Symposium on Theoretical Aspects of Computer Science (STACS), LIPIcs, vol. 47, pp. 31:1–31:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
19. Eiselt, H.A., Gendreau, M., Laporte, G.: Location of facilities on a network subject to a single-edge failure. Networks **22**(3), 231–246 (1992)
20. Feldmann, A.E., Marx, D.: The Parameterized Hardness of the K-Center Problem in Transportation Networks. In: 16Th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden, vol. 101, pp. 19:1–19:13 (2018)
21. Fellows, M.R., Fernau, H.: Facility location problems: a parameterized view. Discret. Appl. Math. **159**(11), 1118–1130 (2011)
22. Fomin, F.V., Lokshtanov, D., Raman, V., Saurabh, S.: Subexponential algorithms for partial cover problems. Inf. Process. Lett. **111**(16), 814–818 (2011)
23. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 503–510. SIAM (2010)
24. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Kernels for (connected) dominating set on graphs with excluded topological minors. ACM Trans. Algorithms **14**(1), 6:1–6:31 (2018)
25. Frick, M., Grohe, M.: Deciding First-Order Properties of Locally Tree-Decomposalbe Graphs. In: Automata, Languages and Programming, 26Th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings, pp. 331–340 (1999)
26. Hassin, R., Ravi, R., Salman, F.S.: Tractable Cases of Facility Location on a Network with a Linear Reliability Order of Links, pp. 275–276. Springer, Berlin (2009)
27. Hassin, R., Ravi, R., Salman, F.S.: Multiple facility location on a network with linear reliability order of edges. J. Comb. Optim. 1–25 (2017)
28. Hochbaum, D.S. (ed.): Approximation Algorithms for NP-hard Problems. PWS Publishing Co, Boston (1997)
29. Khuller, S., Moss, A., Naor, J.: The budgeted maximum coverage problem. Inf. Process. Lett. **70**(1), 39–45 (1999)
30. Kloks, T.: Treewidth, Computations and Approximations, Lecture Notes in Computer Science, vol. 842. Springer (1994)
31. Kneis, J., Mölle, D., Rossmanith, P.: Partial vs. complete domination: T-dominating set. In: Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM '07, pp. 367–376. Springer, Berlin (2007)
32. Melachrinoudis, E., Helander, M.E.: A single facility location problem on a tree with unreliable edges. Networks **27**(4), 219–237 (1996)
33. Narayanaswamy, N.S., Nasre, M., Vijayaragunathan, R.: Facility Location on Planar Graphs with Unreliable Links. In: Computer Science - Theory and Applications - 13Th International Computer Science Symposium in Russia, CSR 2018, Moscow, Russia, June 6-10, 2018, Proceedings, pp. 269–281 (2018)
34. Philip, G., Raman, V., Sikdar, S.: Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. ACM Trans. Algorithms **9**(1), 11 (2012)